Recent Developments in Probabilistic Circuits

Guy Van den Broeck

Google - Feb 9, 2022
Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
The Alphabet Soup of probabilistic models


Intractable and tractable models

- Fully factorized
- NaiveBayes
- AndOrGraphs
- PDGs
- Trees
- PSDDs
- CNets
- LTM s
- SPNs
- NADEs
- Thin Junction Trees
- ACs
- MADEs
- MAFs
- VAEs
- DPPs
- FVSBNs
- TACs
- IAFs
- NAFs
- RAEs
- Mixtures
- BNs
- NICE
- FGs
- GANs
- RealNVP
- MNs
"Every talk needs a joke and a literature overview slide, not necessarily distinct"
- after Ron Graham
a *unifying framework* for tractable models
Probabilistic circuits

_computational graphs_ that recursively define distributions

\[ p(X_1) = w_1 p_1(X_1) + w_2 p_2(X_1) \]

⇒ _mixtures_

\[ p(X_1, X_2) = p(X_1) \cdot p(X_2) \]

⇒ _factorizations_

\[ X_3 \quad X_4 \]

\[ X_3 \quad X_4 \]
Likelihood  \[ p(X_1 = -1.85, X_2 = 0.5, X_3 = -1.3, X_4 = 0.2) \]
Likelihood

\[ p(X_1 = -1.85, X_2 = 0.5, X_3 = -1.3, X_4 = 0.2) \]
Likelihood

\[ p(X_1 = -1.85, X_2 = 0.5, X_3 = -1.3, X_4 = 0.2) \]
A sum node is **smooth** if its children depend on the same set of variables.

A product node is **decomposable** if its children depend on disjoint sets of variables.

**Tractable marginals**

Darwiche and Marquis, “A Knowledge Compilation Map”, 2002
Smoothness + decomposability = tractable MAR

If \( p(x) = \sum_i w_i p_i(x) \), (smoothness):

\[
\int p(x) \, dx = \int \sum_i w_i p_i(x) \, dx = \sum_i w_i \int p_i(x) \, dx
\]

\( \implies \) integrals are “pushed down” to children

[Darwiche & Marquis JAIR 2001, Poon & Domingos UAI11]
\[ p(x, y, z) = p(x)p(y)p(z), \text{ (decomposability)}: \]
\[
\int \int \int p(x, y, z) \, dx \, dy \, dz = \\
= \int \int \int p(x)p(y)p(z) \, dx \, dy \, dz = \\
= \int p(x) \, dx \int p(y) \, dy \int p(z) \, dz
\]

\[ \Rightarrow \text{ integrals decompose into easier ones} \]
**Smoothness** + **decomposability** = **tractable MAR**

Forward pass evaluation for MAR

\[ \Rightarrow \text{linear in circuit size!} \]

E.g. to compute \( p(x_2, x_4) \):

- leafs over \( X_1 \) and \( X_3 \) output \( Z_i = \int p(x_i) \, dx_i \)

\[ \Rightarrow \text{for normalized leaf distributions: } 1.0 \]

- leafs over \( X_2 \) and \( X_4 \) output **EVI**

- feedforward evaluation (bottom-up)
Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
Lossless Data Compression

Data → Encode → Bitstream → Decode → Reconstructed data

Expressive probabilistic model $p(x)$

+ Efficient coding algorithm

Determines the theoretical limit of compression rate

How close we can approach the theoretical limit

Learning Expressive Probabilistic Circuits

Hidden Chow-Liu Trees: CLT-based latent variable PGM/PC

Learned CLT structure captures strong pairwise dependencies

Compile into an equivalent PC

Mini-batch Stochastic Expectation Maximization

A Typical Streaming Code – Arithmetic Coding

We want to compress a set of variables (e.g., pixels, letters) \( \{x_1, x_2, \ldots, x_k\} \)

Compress \( x_1 \) with \(- \log p(x_1)\) bits

Compress \( x_2 \) with \(- \log p(x_2|x_1)\) bits

Compress \( x_3 \) with \(- \log p(x_3|x_1, x_2)\) bits

Need to compute

\[
\begin{align*}
    p(X_1 < x_1) \\
    p(X_1 \leq x_1) \\
    p(X_2 < x_2|x_1) \\
    p(X_2 \leq x_2|x_1) \\
    p(X_3 < x_3|x_1, x_2) \\
    p(X_3 \leq x_3|x_1, x_2) \\
    \vdots
\end{align*}
\]
Efficient Lossless Compression

Need to compute
\[
\begin{align*}
    p(X_1 < x_1) \\
    p(X_1 \leq x_1) \\
    p(X_2 < x_2|x_1) \\
    p(X_2 \leq x_2|x_1) \\
    p(X_3 < x_3|x_1, x_2) \\
    p(X_3 \leq x_3|x_1, x_2) \\
    \vdots
\end{align*}
\]

Fully factorized
- Fast inference
- Not expressive

High tree-width PGMs
- Expressive
- Slow inference

Existing Flow- and VAE-based lossless compression algorithms learn to transform fully factorized distributions into the target distribution.

But en/decoding speed is still relatively slow.
Efficient Lossless Compression with Probabilistic Circuits

Need to compute
\[ p(X_1 < x_1) \]
\[ p(X_1 \leq x_1) \]
\[ p(X_2 < x_2 | x_1) \]
\[ p(X_2 \leq x_2 | x_1) \]
\[ p(X_3 < x_3 | x_1, x_2) \]
\[ p(X_3 \leq x_3 | x_1, x_2) \]
\[ \vdots \]

Fully factorized
- Fast inference
- Not expressive

High tree-width PGMs
- Expressive
- Slow inference

Probabilistic Circuits
- Expressive
- Fast inference

\[ O(\log(D) \cdot |p|) \]
where D is the # variables and |p| is the size of the PC.
Efficient Lossless Compression with Probabilistic Circuits

SoTA compression rates

<table>
<thead>
<tr>
<th>Dataset</th>
<th>HCLT (ours)</th>
<th>IDF</th>
<th>BitSwap</th>
<th>BB-ANS</th>
<th>JPEG2000</th>
<th>WebP</th>
<th>McBits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>1.24 (1.20)</td>
<td>1.96 (1.90)</td>
<td>1.31 (1.27)</td>
<td>1.42 (1.39)</td>
<td>3.37</td>
<td>2.09 (1.98)</td>
<td></td>
</tr>
<tr>
<td>FashionMNIST</td>
<td>3.37 (3.34)</td>
<td>3.50 (3.47)</td>
<td>3.35 (3.28)</td>
<td>3.69 (3.66)</td>
<td>3.93</td>
<td>4.62 (3.72)</td>
<td></td>
</tr>
<tr>
<td>EMNIST (Letter)</td>
<td>1.84 (1.80)</td>
<td>2.02 (1.95)</td>
<td>1.90 (1.84)</td>
<td>2.29 (2.26)</td>
<td>3.62</td>
<td>3.31 (3.12)</td>
<td></td>
</tr>
<tr>
<td>EMNIST (ByClass)</td>
<td>1.89 (1.85)</td>
<td>2.04 (1.98)</td>
<td>1.91 (1.87)</td>
<td>2.24 (2.23)</td>
<td>3.61</td>
<td>3.34 (3.14)</td>
<td></td>
</tr>
</tbody>
</table>

Compress and decompress 5-20x faster than NN methods with similar bitrates

<table>
<thead>
<tr>
<th>Method</th>
<th># parameters</th>
<th>Theoretical bpd</th>
<th>Codeword bpd</th>
<th>Comp. time (s)</th>
<th>Decomp. time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC (HCLT, $M = 16$)</td>
<td>3.3M</td>
<td>1.26</td>
<td>1.30</td>
<td>15</td>
<td>44</td>
</tr>
<tr>
<td>PC (HCLT, $M = 24$)</td>
<td>5.1M</td>
<td>1.22</td>
<td>1.26</td>
<td>26</td>
<td>89</td>
</tr>
<tr>
<td>PC (HCLT, $M = 32$)</td>
<td>7.0M</td>
<td>1.20</td>
<td>1.24</td>
<td>44</td>
<td>155</td>
</tr>
<tr>
<td>IDF</td>
<td>24.1M</td>
<td>1.90</td>
<td>1.96</td>
<td>288</td>
<td>592</td>
</tr>
<tr>
<td>BitSwap</td>
<td>2.8M</td>
<td>1.27</td>
<td>1.31</td>
<td>578</td>
<td>326</td>
</tr>
</tbody>
</table>

Efficient Lossless Compression with Probabilistic Circuits

Can be effectively combined with Flow models to achieve better generative performance

<table>
<thead>
<tr>
<th>Model</th>
<th>CIFAR10</th>
<th>ImageNet32</th>
<th>ImageNet64</th>
</tr>
</thead>
<tbody>
<tr>
<td>RealNVP</td>
<td>3.49</td>
<td>4.28</td>
<td>3.98</td>
</tr>
<tr>
<td>Glow</td>
<td>3.35</td>
<td>4.09</td>
<td>3.81</td>
</tr>
<tr>
<td>IDF</td>
<td>3.32</td>
<td>4.15</td>
<td>3.90</td>
</tr>
<tr>
<td>IDF++</td>
<td><strong>3.24</strong></td>
<td>4.10</td>
<td>3.81</td>
</tr>
<tr>
<td>PC+IDF</td>
<td>3.28</td>
<td><strong>3.99</strong></td>
<td><strong>3.71</strong></td>
</tr>
</tbody>
</table>

Expressive models without *compromises*
Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
Probabilistic circuits seem awfully general.

Are all tractable probabilistic models probabilistic circuits?
Enter: Determinantal Point Processes (DPPs)

DPPs are models where probabilities are specified by (sub)determinants

$$L = \begin{bmatrix} 1 & 0.9 & 0.8 & 0 \\ 0.9 & 0.97 & 0.96 & 0 \\ 0.8 & 0.96 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Tractable likelihoods and marginals
- Global Negative Dependence
- Diversity in recommendation systems

$$\Pr_L(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0) = \frac{1}{\det(L + I)} \det(L_{\{1,2\}})$$
Are all tractable probabilistic models probabilistic circuits?

Probabilistic Circuits (PCs)

Graphical Models (w/ bounded tree-width)

Determinantal Point Processes (DPPs)
Relationship between PCs and DPPs

Probabilistic Circuits

Positive Dependence

Fully Factorized

Determinantal Point Processes

We cannot tractably represent DPPs with subclasses of PCs.

PSDDs

More Tractable

Deterministic and Decomposable PCs

No

Deterministic PCs with no negative parameters

No

Deterministic PCs with negative parameters

No

Decomposable PCs with no negative parameters (SPNs)

No

Decomposable PCs with negative parameters

We don’t know

Fewer Constraints

PCs cannot Tractably Represent DPPs

Theorem 1. For a DPP with kernel $L = B^T \ast B$, where $B$ is randomly generated, with probability 1, this DPP cannot be represented by polynomial-size PSDDs.

Theorem 2. There exists a class of DPPs that cannot be tractably represented by deterministic PCs with (possibly) negative parameters.

Theorem 3. There exists a class of DPPs that cannot be tractably represented by decomposable PCs with non-negative parameters (SPNs).

Probabilistic Generating Circuits

A Tractable Unifying Framework for PCs and DPPs

Probability Generating Functions

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$\Pr_{\beta}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.08</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.12</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.48</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.08</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.04</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.16</td>
</tr>
</tbody>
</table>

$g_{\beta} = 0.16z_1z_2z_3 + 0.04z_1z_2 + 0.08z_1z_3 + 0.02z_1$  
$+ 0.48z_2z_3 + 0.12z_2 + 0.08z_3 + 0.02.$

$g_{\beta} = (0.1(z_1 + 1)(6z_2 + 1) - 0.4z_1z_2)(0.8z_3 + 0.2)$
Probabilistic Generating Circuits (PGCs)

\[ g_\beta = (0.1(z_1 + 1)(6z_2 + 1) - 0.4z_1z_2)(0.8z_3 + 0.2) \]

1. Sum nodes \(\oplus\) with weighted edges to children.
2. Product nodes \(\otimes\) with unweighted edges to children.
3. Leaf nodes: \(z_i\) or constant.

DPPs as PGCs

The generating polynomial for a DPP with kernel $L$ is given by:

$$g_L = \frac{1}{\det(L + I)} \det(I + L\text{diag}(z_1, \ldots, z_n)).$$

- Constant
- Division-free determinant algorithm (Samuelson-Berkowitz algorithm)
- $g_L$ can be represented as a PGC of size $O(n^4)$
PGCs Support Tractable Likelihoods/Marginals

Purely symbolic

\[ z_i = \begin{cases} 
  t, & X_i = 1 \\
  0, & X_i = 0 \\
  1, & \text{otherwise} 
\end{cases} \]

\[ \Pr(X_1 = 1, X_2 = 0, \ldots) = ? \]

\[ p(t) = \alpha_k t^k + \cdots + \alpha_1 t \]

\( \alpha_k \) gives the answer

Example

\[ \text{Pr}(X_2 = 1, X_3 = 0) =? \]

\[
\begin{array}{ccccc|c}
X_1 & X_2 & X_3 & \text{Pr}_\beta \\
\hline
0 & 0 & 0 & 0.02 \\
0 & 0 & 1 & 0.08 \\
0 & 1 & 0 & 0.12 \\
0 & 1 & 1 & 0.48 \\
1 & 0 & 0 & 0.02 \\
1 & 0 & 1 & 0.08 \\
1 & 1 & 0 & 0.04 \\
1 & 1 & 1 & 0.16 \\
\end{array}
\]
# Experiment Results: Amazon Baby Registries

SimplePGC achieves SOTA result on 11/15 datasets

<table>
<thead>
<tr>
<th></th>
<th>DPP</th>
<th>Strudel</th>
<th>EiNet</th>
<th>MT</th>
<th>SimplePGC</th>
</tr>
</thead>
<tbody>
<tr>
<td>bath</td>
<td>-8.55</td>
<td>-8.38</td>
<td>-8.49</td>
<td>-8.53</td>
<td>-8.29*↑o</td>
</tr>
<tr>
<td>bedding</td>
<td>-8.65</td>
<td>-8.50</td>
<td>-8.55</td>
<td>-8.59</td>
<td>-8.41*↑o</td>
</tr>
<tr>
<td>carseats</td>
<td>-4.74</td>
<td>-4.79</td>
<td>-4.72</td>
<td>-4.76</td>
<td>-4.64*↑o</td>
</tr>
<tr>
<td>diaper</td>
<td>-10.61</td>
<td>-9.90</td>
<td>-9.86</td>
<td>-9.93</td>
<td>-9.72*↑o</td>
</tr>
<tr>
<td>feeding</td>
<td>-11.86</td>
<td>-11.42</td>
<td>-11.27</td>
<td>-11.30</td>
<td>-11.17*↑o</td>
</tr>
<tr>
<td>furniture</td>
<td>-4.38</td>
<td>-4.39</td>
<td>-4.38</td>
<td>-4.43</td>
<td>-4.34*↑o</td>
</tr>
<tr>
<td>health</td>
<td>-7.40</td>
<td>-7.37</td>
<td>-7.47</td>
<td>-7.49</td>
<td>-7.24*↑o</td>
</tr>
<tr>
<td>media</td>
<td>-8.36</td>
<td>-7.62</td>
<td>-7.82</td>
<td>-7.93</td>
<td>-7.69↑</td>
</tr>
<tr>
<td>moms</td>
<td>-3.55</td>
<td>-3.52</td>
<td>-3.48</td>
<td>-3.54</td>
<td>-3.53°</td>
</tr>
<tr>
<td>safety</td>
<td>-4.28</td>
<td>-4.43</td>
<td>-4.39</td>
<td>-4.36</td>
<td>-4.28*↑o</td>
</tr>
<tr>
<td>strollers</td>
<td>-5.30</td>
<td>-5.07</td>
<td>-5.07</td>
<td>-5.14</td>
<td>-5.00*↑o</td>
</tr>
<tr>
<td>toys</td>
<td>-8.05</td>
<td>-7.61</td>
<td>-7.84</td>
<td>-7.88</td>
<td>-7.62↑</td>
</tr>
</tbody>
</table>
Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
   Cool things we can do with circuits :-)

Cool things we can do with circuits :-)

Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
   Cool things we can do with circuits :-)

Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
   Cool things we can do with circuits :-)

Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
   Cool things we can do with circuits :-)
Information-theoretic quantities

Variational inference

Compression

Black hole imaging


**Queries as pipelines**

\[ \text{KL}_D(p \parallel q) = \int p(x) \times \log\left(\frac{p(x)}{q(x)}\right) dx \]

Queries as pipelines

\[ H(p, q) = \int p(x) \times \log(q(x)) \, dX \]

\(p\) \(\rightarrow\) \(\times\) \(\rightarrow\) \(\int\)

\(q\) \(\rightarrow\) \(\log\) \(\rightarrow\) \(r\)

\(s\)

⇒ we can reuse the operations!
Determinism

A sum node is *deterministic* if only one of its children outputs non-zero for any input.

\[
\Rightarrow \text{allows tractable MAP inference}
\]

\[\arg\max_x p(x)\]

---

Darwiche and Marquis, “A Knowledge Compilation Map”, 2002
# Tractable circuit operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Tractability</th>
<th>Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUM $\theta_1 p + \theta_2 q$</td>
<td>(+Cmp)</td>
<td>(+SD)</td>
</tr>
<tr>
<td>PRODUCT $p \cdot q$</td>
<td>Cmp (+Det, +SD)</td>
<td>Dec (+Det, +SD)</td>
</tr>
<tr>
<td>POWER $p^n, n \in \mathbb{N}$</td>
<td>SD (+Det)</td>
<td>SD (+Det)</td>
</tr>
<tr>
<td></td>
<td>$p^\alpha, \alpha \in \mathbb{R}$</td>
<td>Sm, Dec, Det (+SD)</td>
</tr>
<tr>
<td>QUOTIENT $p/q$</td>
<td>Cmp; $q$ Det (+p Det, +SD)</td>
<td>Dec (+Det, +SD)</td>
</tr>
<tr>
<td>LOG $\log(p)$</td>
<td>Sm, Dec, Det</td>
<td>Sm, Dec</td>
</tr>
<tr>
<td>EXP $\exp(p)$</td>
<td>linear</td>
<td>SD</td>
</tr>
</tbody>
</table>
**Inference by tractable operations**

Systematically derive tractable inference algorithm of complex queries

<table>
<thead>
<tr>
<th>Query</th>
<th>Tract. Conditions</th>
<th>Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CROSS ENTROPY</strong></td>
<td>Cmp, q Det</td>
<td>#P-hard w/o Det</td>
</tr>
<tr>
<td><strong>SHANNON ENTROPY</strong></td>
<td>Sm, Dec, Det</td>
<td>coNP-hard w/o Det</td>
</tr>
<tr>
<td><strong>RÉNYI ENTROPY</strong></td>
<td>SD</td>
<td>#P-hard w/o SD</td>
</tr>
<tr>
<td><strong>MUTUAL INFORMATION</strong></td>
<td>Sm, SD, Det*</td>
<td>coNP-hard w/o SD</td>
</tr>
<tr>
<td><strong>KULLBACK-LEIBLER DIV.</strong></td>
<td>Cmp, Det</td>
<td>#P-hard w/o Det</td>
</tr>
<tr>
<td><strong>RÉNYI’S ALPHA DIV.</strong></td>
<td>Cmp, q Det</td>
<td>#P-hard w/o Det</td>
</tr>
<tr>
<td><strong>ITAKURA-SAITO DIV.</strong></td>
<td>Cmp, Det</td>
<td>#P-hard w/o Det</td>
</tr>
<tr>
<td><strong>CAUCHY-SCHWARZ DIV.</strong></td>
<td>Cmp</td>
<td>#P-hard w/o Cmp</td>
</tr>
<tr>
<td><strong>SQUARED LOSS</strong></td>
<td>Cmp</td>
<td>#P-hard w/o Cmp</td>
</tr>
</tbody>
</table>

Even harder queries

Marginal MAP

Given a set of query variables $Q \subseteq X$ and evidence $e$,
find: $\arg\max_q p(q|e)$

$\Rightarrow$ i.e. MAP of a marginal distribution on $Q$

\[ NP^{PP}\text{-complete} \] for PGMs

\[ NP\text{-hard} \] even for PCs tractable for marginals, MAP & entropy
Pruning circuits

Any parts of circuit not relevant for MMAP state can be pruned away

e.g. $p(X_1 = 1, X_2 = 0)$

We can find such edges in *linear time*
Iterative MMAP solver

Prune edges

Tighten bounds

<table>
<thead>
<tr>
<th>Dataset</th>
<th>runtime (# solved)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>search (h)</td>
</tr>
<tr>
<td>NLTCS</td>
<td>0.01 (10)</td>
</tr>
<tr>
<td>MSNBC</td>
<td>0.03 (10)</td>
</tr>
<tr>
<td>KDD</td>
<td>0.04 (10)</td>
</tr>
<tr>
<td>Plants</td>
<td>2.95 (10)</td>
</tr>
<tr>
<td>Audio</td>
<td>2041.33 (6)</td>
</tr>
<tr>
<td>Jester</td>
<td>2913.04 (2)</td>
</tr>
<tr>
<td>Netflix</td>
<td>– (0)</td>
</tr>
<tr>
<td>Accidents</td>
<td>109.56 (10)</td>
</tr>
<tr>
<td>Retail</td>
<td>0.06 (10)</td>
</tr>
<tr>
<td>Pumsb-star</td>
<td>2208.27 (7)</td>
</tr>
<tr>
<td>DNA</td>
<td>– (0)</td>
</tr>
<tr>
<td>Kosarek</td>
<td>48.74 (10)</td>
</tr>
<tr>
<td>MSWeb</td>
<td>1543.49 (10)</td>
</tr>
<tr>
<td>Book</td>
<td>– (0)</td>
</tr>
<tr>
<td>EachMovie</td>
<td>– (0)</td>
</tr>
<tr>
<td>WebKB</td>
<td>– (0)</td>
</tr>
<tr>
<td>Reuters-52</td>
<td>– (0)</td>
</tr>
<tr>
<td>20 NewsGrp.</td>
<td>– (0)</td>
</tr>
<tr>
<td>BBC</td>
<td>– (0)</td>
</tr>
<tr>
<td>Ad</td>
<td>– (0)</td>
</tr>
</tbody>
</table>
Tractable Computation of Expected Kernels

- How to compute the expected kernel given two distributions $p, q$?

$$E_{x \sim p, x' \sim q}[k(x, x')]$$

- Circuit representation for kernel functions, e.g.,

$$k(x, x') = \exp \left( - \sum_{i=1}^{4} |X_i - X'_i|^2 \right)$$

Tractable Computation of Expected Kernels: Applications

- Reasoning about support vector regression (SVR) with missing features

\[ \mathbb{E}_{x_m \sim p(X_m | x_0)} \left[ \sum_{i=1}^{m} w_i k(x_i, x) + b \right] \]

missing features

SVR model

- Collapsed Black-box Importance Sampling: minimize kernelized Stein discrepancy

importance weights

\[ w^* = \arg\min_w \left\{ w^T K_{p,s} w \left| \sum_{i=1}^{n} w_i = 1, w_i \geq 0 \right. \right\} \]

expected kernel matrix
As soon as *dice* was put online people started using it in surprising ways we had not foreseen.
Conclusion

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?
tractability is a spectrum
Thanks

This was the work of many wonderful students/postdoc/collaborators!

References: http://starai.cs.ucla.edu/publications/